

# APLIKASI PENCARIAN TITIK LOKASI FASILITAS KESEHATAN TERDEKAT MENGUNAKAN METODE ASYNCHRONOUS PARTICLE SWARM OPTIMIZATION BERBASIS WEBVIEW ANDROID (STUDI KASUS : KOTA BENGKULU)

Rusdi Efendi<sup>1</sup>, Kurnia Anggriani<sup>2</sup>, Elsi Helna Sari<sup>3</sup>

<sup>1</sup>Program Studi Teknik Informatika, <sup>2</sup>Fakultas Teknik, <sup>3</sup>Universitas Bengkulu

<sup>1,2,3</sup>Jl. WR. Supratman Kandang Limun Bengkulu 38371A, telepon (0736) 341022

e-mail : [rusdi.efendi@unib.ac.id](mailto:rusdi.efendi@unib.ac.id)<sup>1</sup>, [kurnia@unib.ac.id](mailto:kurnia@unib.ac.id)<sup>2</sup>, [elsihelna@gmail.com](mailto:elsihelna@gmail.com)<sup>3</sup>

**Abstrak** : Fasilitas kesehatan merupakan tempat pelayanan yang berhubungan dengan kesehatan. Kota Bengkulu menyediakan fasilitas kesehatan guna memberikan pelayanan/sarana bagi masyarakat untuk mengecek kesehatan atau meningkatkan kesehatan. Kondisi nyata yang sering terjadi ketika orang mendadak sakit, dan dia membutuhkan lokasi fasilitas kesehatan terdekat untuk penanganan pertama. Metode *Asynchronous Particle Swarm Optimization* merupakan suatu algoritma yang terinspirasi dari perilaku sekumpulan burung berkelompok untuk mencari makanan. Metode ini merupakan salah satu metode untuk pencarian jarak terpendek. Dengan metode ini, dapat memberikan solusi untuk menghasilkan lokasi terdekat. Dari pengujian yang dilakukan, algoritma *Asynchronous Particle Swarm Optimization* telah berhasil diterapkan pada pencarian titik lokasi fasilitas kesehatan dan telah berhasil dirancang dan dibangun suatu aplikasi pencarian titik lokasi fasilitas kesehatan terdekat berbasis *Webview Android*.

**Kata kunci** : Fasilitas Kesehatan, Pencarian Lokasi Terdekat, *Asynchronous Particle Swarm Optimization*, *Webview Android*.

<sup>1,2</sup> Dosen, Jurusan Teknik Informatika Fakultas Teknik Universitas Bengkulu, Jl. WR. Supratman Kandang Limun Bengkulu 38371A, (telp : 0736-341022), e-mail : [rusdi.efendi@unib.ac.id](mailto:rusdi.efendi@unib.ac.id), [kurnia@unib.ac.id](mailto:kurnia@unib.ac.id)

<sup>3</sup> Mahasiswa, Jurusan Teknik Informatika Fakultas Teknik Universitas Bengkulu, Jl. WR. Supratman Kandang Limun Bengkulu 38371A, (telp : 0736-341022), e-mail : [elsihelna@gmail.com](mailto:elsihelna@gmail.com)

**Abstract** : Health services are health related public services. Bengkulu city provided health services for people to control their health or to improve it. Health services location in Bengkulu city are not placed in the same place therefore makes it difficult to find health services. Other condition is when there urgent issue and we need nearby hospital location. *Asynchronous Particle Swarm Optimization* method is an algorithm inspired by the behavior of a flock of birds flocking in search of food. This

method is one method of finding the shortest distance. With this method, can provide a solution to produce a nearby location. From testing conducted, this research made it possible to implement *Asynchronous Particle Swarm Optimization* algorithm to find nearest health facilities and this research designed and built nearest health facilities finder application with *Webview Android*

**Keywords** : Health Facility, Search Nearby Locations, *Asynchronous Particle Swarm Optimization*, *Web View Android*

## I. PENDAHULUAN

Kesehatan merupakan suatu komponen utama yang paling penting dalam kehidupan. Kesehatan merupakan kesejahteraan seseorang dari segi badan atau fisik. Untuk menjaga kesehatan diperlukan suatu upaya untuk meningkatkan kesehatan. Salah satu upaya peningkatan kesehatan yaitu dengan menjaga pola makan, menjaga pola hidup, dan juga teratur *check up* ke dokter untuk mencegah atau menghindari penyakit. Dokter melakukan praktek di suatu fasilitas kesehatan, seperti rumah sakit, puskesmas, klinik, apotek, maupun praktek mandiri/bersama.

Bertambahnya fasilitas kesehatan menjadi suatu permasalahan yang membuat orang tidak mempunyai informasi tentang fasilitas pelayanan kesehatan terbaru. Kondisi yang sering terjadi saat ini adalah ketika orang sakit, dia tidak dapat memastikan apakah dokter di salah satu fasilitas kesehatan ada atau tidak, apakah fasilitas kesehatan seperti praktek dokter mandiri/bersama buka atau tidak. Dengan kondisi ini, kebanyakan orang mencari informasi melalui telepon, sosial media, atau secara langsung mengunjungi fasilitas kesehatan tersebut. Tentu saja hal ini

tidak efektif, membutuhkan waktu untuk menemukan jawaban/informasi dari hal-hal yang diragukan oleh masyarakat. Tidak hanya itu, kasus yang sering terjadi di kehidupan nyata adalah ketika orang mendadak sakit dan membutuhkan fasilitas pelayanan kesehatan terdekat untuk penanganan pertama. Oleh karena itu diperlukan suatu aplikasi yang dapat memberikan informasi lokasi pelayanan kesehatan terdekat. Adanya aplikasi ini sangat membantu bagi pengguna untuk mempercepat mendapatkan sebuah informasi berupa lokasi dan detailnya.

Penelitian terkait yang telah dilakukan sebelumnya yaitu Hanni Cempaka (2015) tentang Aplikasi Pencarian lokasi Tambal Ban di Kota Bengkulu menggunakan metode Koloni Semut. Aplikasi ini diterapkan pada smartphone berbasis Android. Pencarian lokasi terdekat tambal ban menggunakan metode Koloni semut yang menghasilkan waktu komputasi sebesar 11,4 detik[2]. Penelitian yang dilakukan oleh Ira Prasetyaningrum, Arna Fariza, Alkis Fuady (2012) tentang pencarian jalur terpendek menggunakan Algoritma *Particle Swarm Optimization* (PSO) Di Kabupaten Bangkalan. Dalam penelitian ini, metode PSO ini digunakan untuk mencari jalur terpendek di Kabupaten Bangkalan. Hasil dari uji coba pada penelitian ini adalah pencarian jalur terpendek dengan metode *Particle Swarm Optimization* (PSO) tergantung dari parameter-parameter yang dimasukkan, dan semakin banyak parameter inputan akan menentukan kesuksesan jalur terpendek[4].

Berdasarkan permasalahan dan penelitian yang telah dilakukan sebelumnya, diperlukan suatu metode yang dapat menemukan lokasi terdekat dari fasilitas pelayanan kesehatan. Metode yang digunakan adalah metode *Asynchronous Particle Swarm Optimization* (APSO). APSO digunakan untuk menyelesaikan masalah optimasi.

## II. DASAR TEORI

### 2.1 *Asynchronous Particle Swarm Optimization (APSO)*

#### 1) Particle Swarm Optimization (PSO)

PSO pertama kali diperkenalkan oleh R.C Eberhart dan J.Kennedy pada tahun 1995. Secara sederhana algoritma ini meniru kemampuan binatang yang mencari sumber makanan secara berkelompok. Setiap individu dalam PSO dianggap sebagai sebuah partikel. Menurut R.C Eberhart dan J.Kennedy algoritma ini memiliki keuntungan tersendiri. Salah satunya adalah algoritma ini hanya meliputi algoritma yang sangat sederhana, dan paradigma dapat diimplementasikan hanya dalam beberapa baris kode[3].

Setiap partikel diasumsikan memiliki dua karakter yaitu posisi dan kecepatan. Setiap partikel bergerak dalam

ruang/space tertentu dan mengingat posisi terbaik yang pernah dilalui atau ditemukan terhadap sumber makanan atau nilai fungsi objektif. Setiap partikel menyampaikan informasi atau posisi bagusnya kepada partikel yang lain dan menyesuaikan posisi dan kecepatan masing-masing berdasarkan informasi yang diterima mengenai posisi yang bagus tersebut[5].

PSO memiliki beberapa parameter, yaitu :

1. *Swarm* (jumlah partikel): populasi dari suatu algoritma. Ukuran *swarm* atau populasi yang dipilih adalah pada persoalan yang dihadapi. Pada umumnya, ukuran *swarm* yang digunakan antar 20 sampai 50.
2. Dimensi partikel : Hal ini bergantung pada masalah yang akan di optimasi.
3. *Learning rate* atau laju belajar. Biasanya  $c_1$  dan  $c_2 = 2$ .
4. *Velocity* : *vector* yang menggerakkan proses optimasi yang menentukan arah dimana suatu partikel diperlukan untuk berpindah untuk memperbaiki posisinya semula.
5. Berat inersia : berat inersia disimbolkan dengan  $w$ , parameter ini digunakan untuk mengontrol dampak dari adanya *velocity* yang diberikan suatu partikel. Berat inersia diperkenalkan untuk keseimbangan antara kemampuan penelusuran global dan local. Biasanya  $w=1$ .

Proses pencarian dapat dinyatakan dengan vektor posisi  $X_i^k$ , dimana ( $i = 1,2,3, \dots$ ) dan kecepatan  $V_i^k$ , dimana ( $i = 1,2,3$ ) dalam dimensi ruang pencarian tertentu. Selain itu, optimalitas dari solusi dalam algoritma PSO tergantung pada masing-masing posisi partikel dan memperbarui kecepatan menggunakan persamaan berikut.

$$V_i^{k+1} = w \cdot V_i^k + c_1 \cdot r_1 \cdot (p_i^k - x_i^k) + c_2 \cdot r_2 \cdot (p_g^k - x_i^k) \quad (1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2)$$

dimana :

$i$  = partikel ke  $i$ , dimana  $i = \{1,2,3,4, \dots\}$

$k$  = iterasi PSO ke  $k$ , dimana  $k = \{1,2,3,4, \dots\}$

$w$  = berat inersia

$c_1$  = tingkat akselerasi untuk faktor kognitif

$c_2$  = tingkat akselerasi untuk faktor sosial

$v_i^k$  = kecepatan partikel ke  $i$  pada iterasi ke  $k$

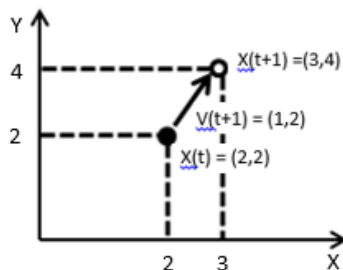
$x_i^k$  = posisi partikel ke  $i$  pada iterasi ke  $k$

$p_i^k$  = local best dari partikel ke  $i$  pada iterasi ke  $k$

$p_g^k$  = global best pada iterasi ke  $k$

$r_1$  dan  $r_2$  = fungsi acak, nilainya antara 0 sampai 1

Menurut Jatmiko, posisi partikel ditentukan oleh koordinat x dan y yang dipresentasikan oleh *vector* titik[3]. Sementara kecepatan partikel juga ditentukan oleh koordinat x dan y yang direpresentasikan dalam bentuk *vector* arah yang dilambangkan dengan v. posisi partikel pada iterasi selanjutnya ditentukan oleh penjumlahan *vector* antara posisi partikel dengan kecepatan partikel. Pada inisialisasi awal, kecepatan dan *local best* masing-masing partikel serta *global best* ditentukan secara acak. Posisi partikel pada iterasi selanjutnya ditentukan oleh penjumlahan *vector* antara posisi partikel dengan kecepatan partikel seperti Gambar. 1.



Gambar. 1 Vektor posisi dan kecepatan partikel

#### 1) Asynchronous Update

Dalam varian ini dari algoritma PSO, pencarian partikel dipandu oleh informasi parsial atau informasi sebagian dari ketetanggaannya atau beberapa informasi tersebut dari iterasi saat ini dan informasi dari iterasi sebelumnya.

Algoritma *Asynchronous Particle Swarm Optimization* pada aplikasi berfungsi untuk menentukan jarak terpendek menuju ke Fasilitas Kesehatan terdekat. Pada kasus ini, Partikel merupakan agen pencari lokasi fasilitas kesehatan. Proses algoritma APSO dimulai ketika pengguna melakukan set posisi. Set posisi ini berupa titik koordinat yang akan menjadi titik awal dari perhitungan APSO. Titik awal ini menjadi posisi awal untuk setiap partikel atau populasi.

Langkah-langkah algoritma APSO adalah sebagai berikut.

1. Inisialisasi parameter konstan  
Inisialisasi jumlah partikel (n), berat inersia (w),  $C_1$  (faktor kognitif),  $C_2$  (faktor sosial), dan maksimal iterasi.
2. Inisialisasi posisi partikel dan batas kecepatan partikel.
3. Inisialisasi nilai fitness pbest dan gbest  
Untuk perhitungan awal, inisialisasi nilai fitness pbest itu sama dengan fitness posisi awal dan nilai gbest dipilih dari nilai fitness terkecil.
4. Memperbarui kecepatan partikel  
Untuk memperbarui kecepatan perlu menentukan Pbest dan Gbest, dan menggunakan inisialisasi parameter konstan. Gbest merupakan posisi terbaik dari seluruh partikel. Partikel dengan jarak total minimum ditetapkan sebagai Gbest. Sedangkan Pbest sama dengan nilai awal

atau posisi awal. Setelah itu dapat menghitung kecepatan dengan menggunakan rumus seperti (1).

#### 5. Memperbarui posisi partikel

Posisi dapat diperbarui dari posisi awal/sebelumnya yang dijumlahkan dengan kecepatan. Populasi baru dapat dihitung menggunakan rumus seperti (2).

#### 6. Evaluasi fitness

Untuk perhitungan selanjutnya didapatkan dari evaluasi jarak yang diperoleh partikel.  $F(x) = \sum d_{ab}$ . Perhitungan jarak menggunakan perhitungan jarak antar dua titik yang mengacu pada jarak antar dua titik di bumi (*latitude* dan *longitude*) dengan persamaan :

$$d_{ab} = 69 \times \sqrt{(lon_a - lon_b)^2 + (lat_a - lat_b)^2} \quad (3)$$

Ket :

$d_{ab}$  = jarak antara dua titik koordinat

$lon_a$  = titik *longitude* a

$lon_b$  = titik *longitude* b

$lat_a$  = titik *latitude* a

$lat_b$  = titik *latitude* b

#### 7. Langkah selanjutnya melakukan pengecekan apakah nilai fitness lebih baik dari pbest.

Jika “ya” set fitness sebagai pbest yang baru

Jika “tidak” maka lanjut ke langkah 8.

#### 8. Melakukan pengecekan apakah nilai fitness lebih baik dari gbest.

Jika “ya” set fitness sebagai gbest yang baru

Jika “tidak” maka lanjut ke langkah 9.

#### 9. Melakukan pengecekan apakah mencapai maksimal iterasi

Jika “ya” maka pencarian dihentikan dan lanjut ke langkah 10.

Jika “tidak” maka lanjutkan pencarian dan kembali ke langkah 4.

#### 10. Solusi ditemukan dan selesai.

Solusi yang ditemukan merupakan lokasi terdekat dengan jarak terkecil.

Diagram alur algoritma *Asynchronous Particle Swarm Optimization* yang ditunjukkan pada Gambar.3.

#### 2) Perbedaan Synchronous dan Asynchronous

Pada *synchronous update*, semua partikel digerakkan secara paralel kemudian partikel terbaik di dalam topologi ketetanggaan dipilih dan iterasi berikutnya dijalankan. Sedangkan pada *asynchronous update*, partikel terbaik di dalam topologi ketetanggaan dipilih lebih dulu, kemudian partikel terbaik tersebut diperhitungkan untuk menggerakkan semua partikel lainnya. Menurut Carlisle, *asynchronous update* lebih efisien dibandingkan *synchronous update*.

Dalam PSO asli, *local best* partikel ( $P_i$ ) dan *global best* ( $P_g$ ) diperbarui setelah semua fitness partikel dievaluasi. Hal

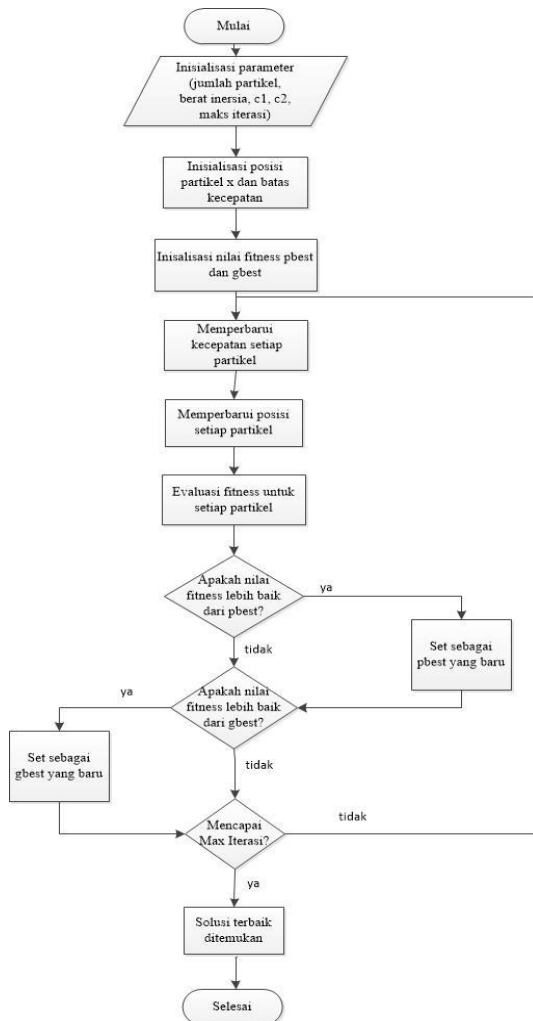
ini dikenal sebagai pembaruan *Synchronous Update*[1]. *Synchronous Update* memastikan semua partikel, informasi yang sempurna dan lengkap dari tetangganya. *Synchronous Update* adalah pilihan yang mahal, sebagai sebuah partikel harus menunggu seluruh kawanan diperbarui sebelum dapat pindah ke posisi baru dan melanjutkan pencariannya[1]. Oleh karena itu partikel pertama dievaluasi untuk waktu yang lama karena menunggu seluruh kawanan diperbarui. Algoritma SPSO dapat dilihat pada Gambar.2.

```

Initialize particles population;
Do{
    Calculate fitness values of each particles using fitness function;
    Update  $P_{id}$  (if the current fitness value is better than  $P_{id}$ );
    Determine  $P_{gd}$  (choose the particle position with the best fitness value of all the neighbours as the  $P_{gd}$ );
    For each particle {
        Update velocity and position;
    }
} while maximum iteration or ideal fitness is not attained;

```

Gambar. 2 Synchronous PSO



Gambar. 3 Flowchart APSO

Dalam APSO, *local best* ( $P_i$ ), *global best* ( $P_g$ ), kecepatan partikel dan posisi diperbarui setelah fitness dievaluasi. Dalam versi ini, pencarian partikel dipandu oleh informasi parsial atau tidak sempurna dari ketetanggaannya atau beberapa informasi tersebut dari iterasi saat ini dan informasi dari iterasi sebelumnya[1]. Hal ini memberikan kontribusi pada keanekaragaman dalam kawanan, yang merupakan sifat yang diinginkan. Algoritma APSO dapat dilihat pada Gambar.4.

```

Initialize particles population;
Do{
    For each particle {
        Calculate particle's fitness values using fitness function;
        Update  $P_{id}$  (if the current fitness value is better than  $P_{id}$ );
        Update  $P_{gd}$  (if the current fitness value is better than  $P_{gd}$ );
        Update velocity and position;
    }
} while maximum iteration or ideal fitness is not attained;

```

Gambar. 4 Asynchronous PSO

Berikut merupakan tabel perbedaan Synchronous dan Asynchronous.

Tabel. 1 Perbedaan SPSO dan APSO

	SPSO	APSO
<b>Update</b>	Pada awal setiap iterasi memperbarui semua partikel	Memperbarui partikel setelah fitness dievaluasi
<b>Informasi</b>	Semua partikel menggunakan informasi dari iterasi yang sama	Partikel menggunakan kombinasi informasi dari iterasi saat ini dan iterasi sebelumnya
<b>gBest</b>	Semua partikel memperbarui berdasarkan yang terbaik dari iterasi sebelumnya	Partikel menggunakan informasi yang tersedia saat ini, keragaman yang lebih baik

#### 4) Kondisi Pemberhentian Algoritma PSO

Terdapat beberapa kondisi yang dapat dijadikan parameter untuk membuat PSO berhenti[1].

1. Iterasi yang terjadi sudah melebihi batas waktu yang ditentukan
2. Solusi yang bisa diterima sudah ditemukan
3. Tidak ada peningkatan dalam beberapa iterasi
4. Radius normal dari kumpulan partikel mendekati nol.  
Radius dapat dihitung sebagai berikut.

$$R_{norm} = \frac{R_{max}}{\text{diameter}(S)} \quad (4)$$

Ket :

Diameter(S) : diameter dari kumpulan partikel awal

$r_{max}$  : jarak maksimum partikel dari *global best*.

5. Ketika perubahan solusi yang ditemukan mendekati nol.

### III. METODE PENELITIAN

#### 3.1 Jenis Penelitian

Jenis penelitian yang akan dilakukan adalah penelitian terapan (*applied research*). Pada penelitian ini, peneliti membuat aplikasi yang mampu menampilkan hasil pencarian titik lokasi fasilitas kesehatan terdekat dengan menerapkan metode *Asynchronous Particle Swarm Optimization*.

#### 3.2 Subjek Penelitian

- 1) Populasi : Populasi yang digunakan dalam penelitian ini adalah objek berupa fasilitas kesehatan yang terdapat di Kota Bengkulu.
- 2) Sampel : Sampel pada penelitian ini adalah 8 rumah sakit, 20 puskesmas, 18 klinik, 27 praktek mandiri/perorangan atau bersama, dan 40 apotek di Kota Bengkulu.
- 3) Sampling : Pengambilan sampel yang dilakukan difokuskan terhadap 113 fasilitas kesehatan di Kota Bengkulu yang diambil sendiri dengan menggunakan GPS Garmin dan aplikasi GPS yang terdapat pada smartphone milik pribadi dan orang lain.

#### 3.3 Sarana Pendukung

Dalam pembuatan aplikasi dalam penelitian ini diperlukan beberapa perangkat lunak dan perangkat keras yang membantu penyelesaian aplikasi.

##### a. Perangkat Lunak

Berikut ini merupakan perangkat lunak yang diperlukan :

1. *Sistem Operasi Windows 8*
2. *Codeigniter*
3. *Notepad++*
4. *Android Studio*
5. *Google Maps*
6. *Microsoft Office Word 2013*
7. *Microsoft Office Visio 2010*
8. *Astah Community*

##### b. Perangkat Keras

Berikut ini merupakan perangkat keras yang diperlukan :

1. *Monitor*
2. *Processor Intel Core i7*
3. *RAM 2 GB*
4. *Keyboard*
5. *Mouse*
6. *Harddisk 1 TB*

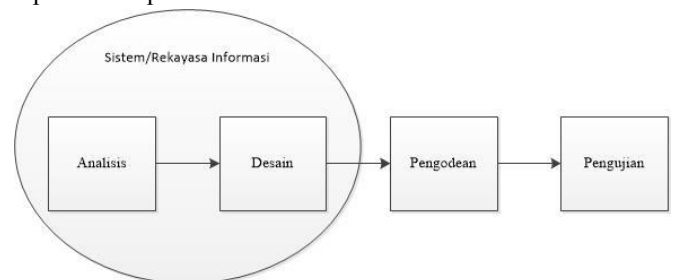
#### 3.4 Teknik Pengumpulan Data

Teknik yang digunakan pada penelitian ini adalah sebagai berikut :

1. Teknik Pustaka
2. Teknik Observasi
3. Teknik Wawancara

#### 3.5 Metode Pengembangan Sistem

Metode pengembangan yang akan digunakan dalam merancang aplikasi pencarian lokasi terdekat ini adalah model *waterfall* (*air terjun*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung. Berikut adalah model *waterfall* yang dapat dilihat pada Gambar.5.



Gambar. 5 Ilustrasi Model Waterfall

#### 3.6 Metode Pengujian

Metode pengujian yang digunakan pada aplikasi ini adalah metode kuantitatif.

##### a) White Box Testing

Pengujian white box bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak apakah dapat berjalan dengan baik atau tidak. Metode yang digunakan pada white box adalah metode *basis path*.

##### b) Black Box Testing

Metode yang digunakan pada *black box* adalah metode *equivalen partitioning testing* yang melakukan pengujian dengan membagi domain input dari suatu program ke dalam

kelas data, menentukan kasus pengujian dengan mengungkapkan kelas-kelas kesalahan.

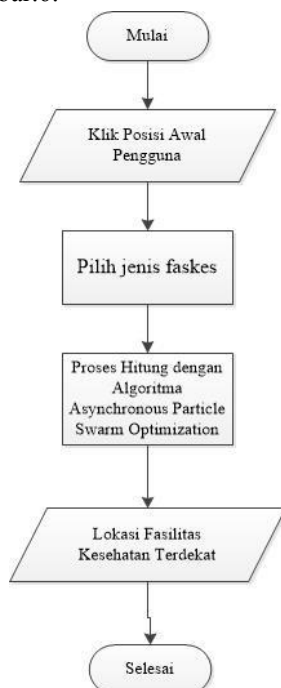
#### c) Pengujian Parameter APSO

Pengujian ini untuk mengetahui parameter mana yang dapat digunakan untuk mendapatkan hasil pencarian yang optimal dengan waktu komputasi tersingkat

### IV. ANALISA DAN PERANCANGAN

#### 4.1 Analisis Cara Kerja Sistem

Penerapan algoritma *Asynchronous Particle Swarm Optimization* pada aplikasi ini adalah untuk menentukan jalur terpendek dari posisi pengguna menuju lokasi fasilitas kesehatan di Kota Bengkulu. Diagram alur kerja sistem dapat dilihat pada Gambar.6.



Gambar. 6 Alur Kerja Sistem

#### 4.2 Analisis Fungsional

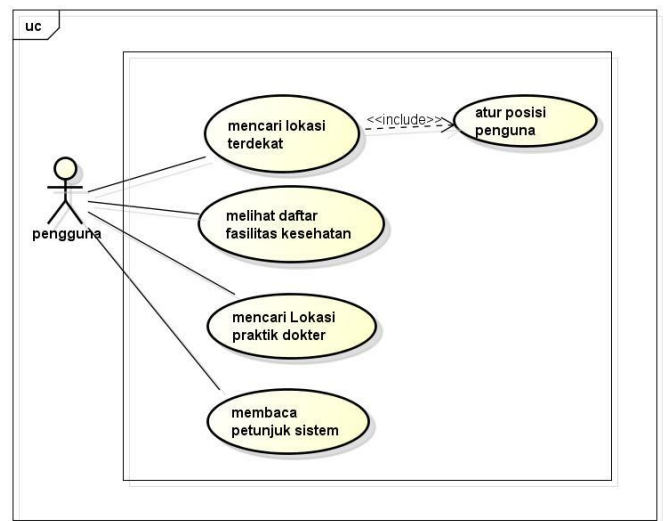
Berikut merupakan beberapa fitur yang ada pada sistem yang akan dibangun.

- 1) Aplikasi harus dapat menampilkan posisi pengguna pada peta.
  - Pengguna dapat melihat posisi keberadaannya
- 2) Aplikasi harus dapat menampilkan lokasi terdekat dan jalur terpendek menuju fasilitas kesehatan terdekat dari hasil perhitungan dengan menggunakan Algoritma *Asynchronous Particle Swarm Optimization*.
  - Pengguna dapat melihat lokasi terdekat dan rute terpendek menuju fasilitas kesehatan

- Pengguna dapat melihat informasi dari fasilitas kesehatan terdekat
- 3) Aplikasi harus dapat menampilkan daftar fasilitas kesehatan berdasarkan jenis.
    - Pengguna dapat melihat daftar nama lokasi berdasarkan jenis masing-masing
    - pengguna dapat melihat lokasi fasilitas kesehatan
    - Pengguna dapat melihat detail informasi setiap fasilitas kesehatan.
  - 4) Aplikasi harus dapat menampilkan lokasi praktisi dokter.
    - Pengguna dapat melihat lokasi berdasarkan spesialisasi dokter.
    - Pengguna dapat melihat lokasi berdasarkan nama dokter
    - Pengguna dapat melihat detail informasi dari lokasi fasilitas kesehatan.
    - Pengguna dapat melihat dokter yang buka dengan jam sekarang
  - 5) Aplikasi harus dapat menampilkan petunjuk
    - pengguna dapat melihat informasi petunjuk aplikasi

#### 4.3 Perancangan Use Case Diagram

Berikut merupakan *use case diagram* dari sistem yang akan ditunjukkan pada Gambar.7.



Gambar. 7 Use Case Diagram

Dilihat dari Gambar.7, sistem ini terdapat satu aktor yaitu pengguna. Pengguna pada sistem ini merupakan masyarakat yang ingin mencari lokasi fasilitas kesehatan lokasi.



## V. PEMBAHASAN

### 5.1 Implementasi Antarmuka

#### 1. Halaman Utama

Berikut merupakan tampilan *webview android* dari halaman utama yang akan dijelaskan pada Gambar.8.



Gambar. 8 Halaman Utama Webview Android

Gambar.8 diatas, merupakan halaman utama ketika dibuka melalui Android. Tampilan pada Android menyesuaikan dengan ukuran Android masing-masing. Halaman utama nya terdapat 3 menu yaitu lokasi terdekat, daftar fasilitas kesehatan, dan lokasi praktik dokter.

#### 2. Halaman Lokasi Terdekat

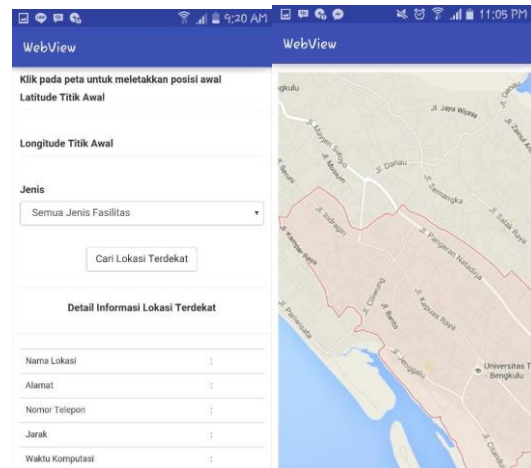
Ketika pengguna memilih menu lokasi terdekat, maka sistem akan menampilkan lokasi terdekat per kecamatan. Berikut tampilan lokasi terdekat per kecamatan yang dilihat dari Android yang akan ditunjukkan pada Gambar.9.



Gambar. 9 Lokasi Terdekat Webview Android

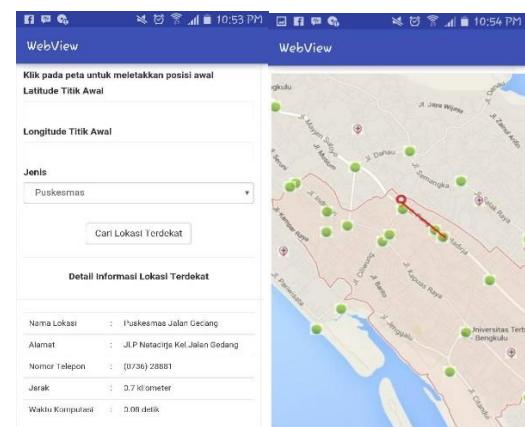
Pada Gambar.9, terdapat 9 kecamatan yang ada di Kota Bengkulu. Untuk tampilan di Android pengguna bisa *scroll* ke bawah untuk melihat kecamatan lainnya. Jika pengguna memilih salah satu kecamatan, misalnya Ratu Samban, maka aplikasi akan menampilkan seperti Gambar.10.

Gambar.10 merupakan tampilan lokasi terdekat ketika di buka di Android. Untuk inputan terletak pada bagian atas, dan untuk detail informasi terletak pada bagian bawah. Posisi inputan dan detail informasi menyesuaikan ukuran Android. Pada menu ini pengguna bisa *scroll* ke bawah untuk melihat peta nya yang dapat dilihat seperti gambar sebelah kanan.



Gambar. 10 Lokasi Terdekat Webview Android (pilih kecamatan)

Ketika pengguna mengklik posisi awal dan memilih jenis fasilitas kesehatan, kemudian klik tombol cari lokasi terdekat, maka aplikasi akan menampilkan tampilan seperti gambar berikut.



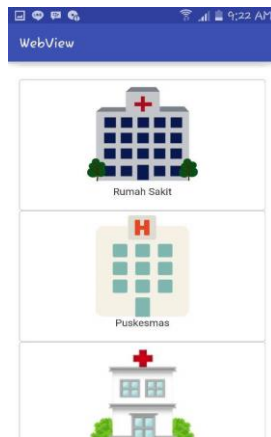
Gambar. 11 Hasil Lokasi Terdekat Webview Android

Pada Gambar.11 merupakan tampilan hasil lokasi terdekat dengan detail informasi berupa nama lokasi, alamat, nomor telepon, jarak, dan waktu komputasi pada bagian atas. Pada peta menunjukkan lokasi awal dan lokasi tujuan (lokasi terdekat). Untuk lebih detail menu ini bisa di *scroll* ke bawah.

#### 3. Halaman Daftar Fasilitas Kesehatan

Tampilan menu daftar fasilitas kesehatan ketika dibuka melalui Android dapat dilihat seperti Gambar.12. Gambar.12 merupakan tampilan menu daftar fasilitas kesehatan. Pada

halaman ini pengguna dapat *scroll* ke bawah untuk melihat berbagai jenis fasilitas kesehatan yaitu rumah sakit, puskesmas, klinik, praktik mandiri perorangan/bersama, dan apotek.

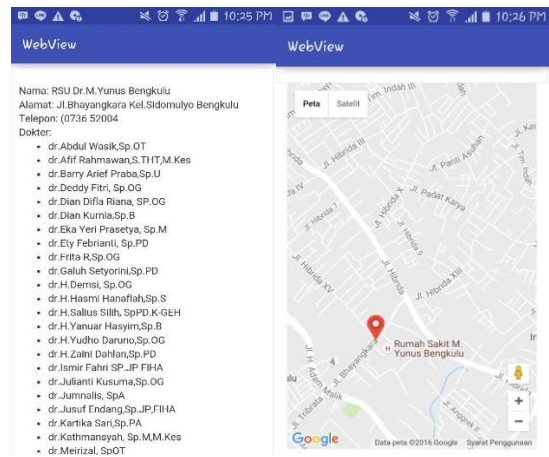


Gambar. 12 Menu Daftar Faskes Webview Android

Ketika pengguna mengklik salah satu jenis fasilitas kesehatan, maka aplikasi akan menampilkan seperti Gambar.13. Gambar.13 merupakan tampilan yang menampilkan seluruh daftar rumah sakit yang ada di Kota Bengkulu. Untuk melihat lokasinya, pengguna dapat mengklik *link* “lihat lokasi” pada detail. Ketika klik “lihat lokasi” maka aplikasi akan menampilkan seperti Gambar.14.

NO	Nama Faskes	Jenis Faskes	Detail
1	RS Bhayangkara Jitra	Rumah Sakit	Lihat Lokasi
2	RS DKT Bengkulu	Rumah Sakit	Lihat Lokasi
3	RS Raflesia	Rumah Sakit	Lihat Lokasi
4	RS Tiara Sella	Rumah Sakit	Lihat Lokasi
5	RSJKO Soeprapto	Rumah Sakit	Lihat Lokasi
6	RSU Dr.M.Yunus Bengkulu	Rumah Sakit	Lihat Lokasi
7	RSU Ummi Bengkulu	Rumah Sakit	Lihat Lokasi
8	RSUD Kota Benokulu	Rumah Sakit	Lihat Lokasi

Gambar. 13 Menu Daftar Faskes Webview Android (jenis)

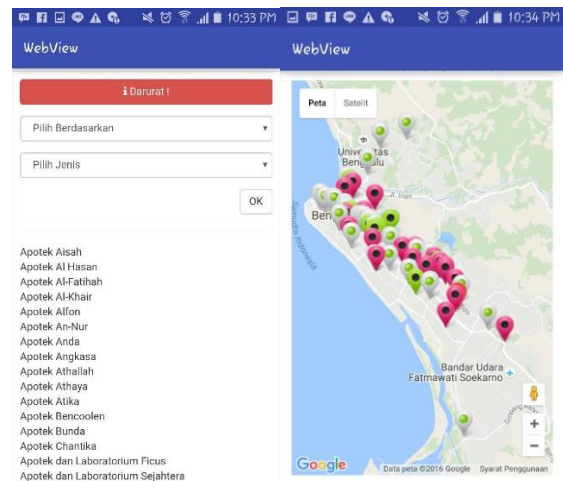


Gambar. 14 Menu Daftar Faskes Webview Android (Detail)

Pada Gambar.14 diatas, aplikasi menampilkan lokasi salah satu rumah sakit yaitu Rumah Sakit M.Yunus pada peta terlihat pada gambar sebelah kanan. Pada tampilan terdapat detail informasi berupa nama rumah sakit, alamat, dan telepon, dan nama dokter yang ada di rumah sakit.

#### 4. Halaman Lokasi Praktik Dokter

Halaman lokasi praktik dokter ketika di buka di Android akan menampilkan tampilan seperti Gambar.15. Pada tampilan seperti Gambar.15, pengguna dapat melakukan beberapa aksi, yaitu pengguna dapat mengklik tombol darurat yang berfungsi untuk mengetahui lokasi praktek dokter yang buka saat ini. Berikut merupakan tampilan ketika pengguna mengklik tombol “darurat” yang akan dijelaskan pada Gambar.16.



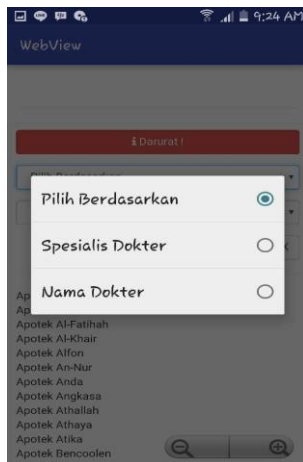
Gambar. 15 Menu Lokasi Praktik Dokter Webview Android





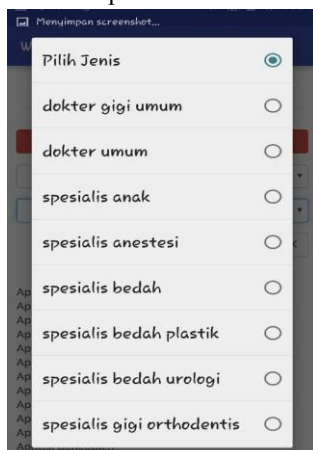
Gambar. 16 Tampilan untuk Dokter yang Buka Saat Ini

Jika pengguna tidak ingin mencari dokter yang buka saat ini, pengguna dapat langsung memilih pilihan cari lokasi dokter berdasarkan spesialisasi dokter atau nama dokter yang akan ditunjukkan pada Gambar.17. Pada Gambar.17, pengguna dapat mencari lokasi praktik dokter berdasarkan spesialisasi dokter.



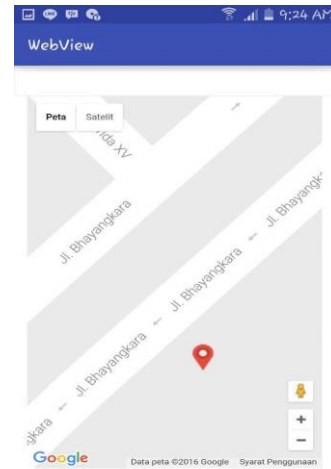
Gambar. 17 Tampilan Pilihan Cari Lokasi Dokter

Ketika klik spesialisasi dokter, maka pengguna akan memilih jenis spesialisasi dokter seperti Gambar.18.



Gambar. 18 Pilih Jenis Spesialisasi Dokter

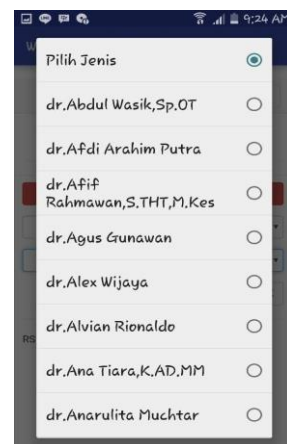
Dapat dilihat pada Gambar.18, ada beberapa spesialisasi dokter yang dapat dipilih pengguna sesuai dengan keinginan dan kebutuhan pengguna. Misalnya, pengguna memilih spesialis anastesi, maka aplikasi akan menampilkan tampilan seperti Gambar.19.



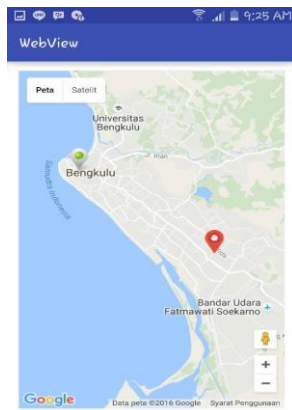
Gambar. 19 Hasil Pencarian Lokasi Spesialisasi Dokter

Gambar.19 menampilkan lokasi praktik dokter spesialisasi anastesi, yang berada di Rumah Sakit M.Yunus. Pada tampilan ini terdapat detail informasi yang dapat di scroll ke atas.

Jika pengguna memilih mencari lokasi praktik dokter berdasarkan nama, maka pengguna harus memilih nama dokter yang ingin ia cari. Pilihan nama dokter akan ditunjukkan pada Gambar.20. Dapat dilihat pada Gambar.20, terdapat nama-nama dokter yang dapat dipilih pengguna. Misalnya, jika pengguna memilih salah satu nama dokter, maka aplikasi akan melakukan pencarian lokasi berdasarkan nama dokter. Tampilan hasil pencarian dapat ditunjukkan pada Gambar.21.



Gambar. 20 Pilih Nama Dokter

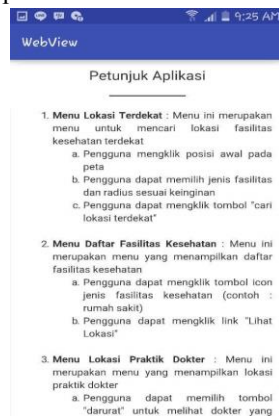


Gambar. 21 Hasil Pencarian Lokasi Nama Dokter

Pada Gambar.21, aplikasi menampilkan lokasi praktek dokter berdasarkan nama dokter yang ditampilkan pada peta. Pada peta terdapat 2 lokasi dimana dokter yang terpilih praktik di 2 tempat tersebut.

## 5. Halaman Petunjuk

Halaman Petunjuk berguna untuk memberikan informasi tentang penggunaan setiap menu yang ada pada aplikasi pencarian lokasi fasilitas kesehatan. Berikut merupakan tampilan halaman petunjuk ketika di buka di Android yang akan ditunjukkan pada Gambar.22.



Gambar. 22 Halaman Petunjuk Webview Android

### 5.2 Pengujian Parameter Algoritma APSO

Pengujian dilakukan dengan mengubah parameter jumlah partikel ( $n$ ), berat inersia ( $w$ ),  $C_1$ , dan  $C_2$  terhadap waktu komputasi dengan nilai yang bervariasi, sehingga didapatkan waktu komputasi untuk menemukan solusi terbaik. Berikut nilai parameter yang akan diujikan yang akan ditunjukkan pada Tabel.2.

Tabel. 2 Daftar Nilai Uji Pada Parameter APSO

Parameter	Nilai Uji
Jumlah Partikel	{10,50,100}
Berat inersia	{0,1,0,2,0,3,...,1}
$C_1$	{0,5,1,1,5,2}
$C_2$	{0,5,1,1,5,2}

Hasil pengujian kinerja algoritma dengan percobaan perubahan nilai parameter jumlah partikel ( $n$ ) terhadap lamanya proses pencarian (waktu komputasi) yang dipilih dapat dilihat pada Tabel.3.

Tabel. 3 Pengujian Jumlah Parameter Terhadap Waktu Komputasi (10 Partikel)

No	Jumlah Partikel	Berat inersia	$C_1$	$C_2$	Waktu Komputasi
1	10	1	2	2	0.2 detik
2	10	1	2	2	0.21 detik
3	10	1	2	2	0.19 detik
4	10	1	2	2	0.16 detik
5	10	1	2	2	0.18 detik
6	50	1	2	2	0.73 detik
7	50	1	2	2	0.66 detik
8	50	1	2	2	0.63 detik
9	50	1	2	2	0.61 detik
10	50	1	2	2	0.49 detik
11	100	1	2	2	1.14 detik
12	100	1	2	2	1.17 detik
13	100	1	2	2	1.1 detik
14	100	1	2	2	1.06 detik
15	100	1	2	2	1.22 detik

Dapat dilihat pada Tbl.3, dari 15 kali percobaan yang dilakukan, dengan jumlah partikel =10, didapatkan waktu komputasi tersingkat yaitu 0,16 detik. Percobaan dengan jumlah partikel = 50, didapatkan waktu komputasi tersingkat yaitu 0,61 detik. Percobaan dengan jumlah partikel = 100, didapatkan waktu komputasi tersingkat yaitu 1,06 detik.

Menurut Carlisle, jumlah partikel tidak terlalu berpengaruh terhadap solusi optimum yang dihasilkan PSO, tetapi berpengaruh terhadap kecepatan proses[6]. Hal ini terbukti dari pengujian jumlah partikel yang diubah-ubah yaitu 10,50, dan 100, bahwa semakin kecil jumlah partikel maka waktu komputasinya akan semakin cepat. Sebaliknya, jika jumlah partikel besar maka waktu komputasinya akan lebih lama.

Tabel. 4 Hasil Pengujian Parameter W Terhadap Waktu Komputasi

No	Jumlah Partikel	Berat inersia	$C_1$	$C_2$	Waktu Komputasi
1	10	1	2	2	0.21 detik
2	10	0.9	2	2	0.17 detik
3	10	0.8	2	2	0.14 detik
4	10	0.7	2	2	0.14 detik
5	10	0.6	2	2	0.17 detik
6	10	0.5	2	2	0.15 detik
7	10	0.4	2	2	0.18 detik
8	10	0.3	2	2	0.14 detik
9	10	0.2	2	2	0.16 detik
10	10	0.1	2	2	0.14 detik

Dapat dilihat dari Tabel.4, dari 10 kali percobaan, dengan jumlah partikel 10 dan  $w$  yang diturunkan dari 1

hingga 0,1, didapatkan hasil waktu komputasi/lamanya proses pencarian yang bervariasi pula yaitu dari 0,14 detik sampai 0,21 detik.

Tabel. 5 Hasil Pengujian W Terhadap Waktu Komputasi (Partikel 50)

No	Jumlah Partikel	Berat inersia	C1	C2	Waktu Komputasi
1	50	1	2	2	0.55 detik
2	50	0.9	2	2	0.52 detik
3	50	0.8	2	2	0.48 detik
4	50	0.7	2	2	0.47 detik
5	50	0.6	2	2	0.49 detik
6	50	0.5	2	2	0.39 detik
7	50	0.4	2	2	0.4 detik
8	50	0.3	2	2	0.46 detik
9	50	0.2	2	2	0.36 detik
10	50	0.1	2	2	0.41 detik

Dapat dilihat pada Tabel.5, dari 10 kali percobaan, dengan jumlah partikel 50, dan w diturunkan dari 1 hingga 0,1, didapatkan hasil waktu komputasi yang bervariasi, dari yang terkecil  $w = 0,2$  dengan waktu komputasi 0,36 detik dan terbesar  $w = 1$  dengan waktu komputasi 0,55 detik.

Tabel. 6 Hasil Pengujian W Terhadap Waktu Komputasi (Partikel 100)

No	Jumlah Partikel	Berat inersia	C1	C2	Waktu Komputasi
1	100	1	2	2	1.1 detik
2	100	0.9	2	2	0.93 detik
3	100	0.8	2	2	1.02 detik
4	100	0.7	2	2	0.87 detik
5	100	0.6	2	2	0.2 detik
6	100	0.5	2	2	0.71 detik
7	100	0.4	2	2	0.8 detik
8	100	0.3	2	2	0.74 detik
9	100	0.2	2	2	0.75 detik
10	100	0.1	2	2	0.77 detik

Dapat dilihat pada Tabel.6, dari 10 kali percobaan, dengan jumlah partikel 100, dan w diturunkan dari 1 hingga 0,1, didapatkan hasil waktu komputasi yang bervariasi. Percobaan dengan waktu tersingkat dengan  $w = 0,6$  dengan waktu 0,2 detik dan terbesar dengan  $w = 1$  dengan waktu 1,1 detik.

Menurut Y.Shi dan Russ Eberhart, penggunaan w berpengaruh untuk meningkatkan performansi PSO[6]. Hal ini terbukti, dari hasil lokasi terdekat yaitu Klinik Asy-Syifa dengan jarak 0,51 km. Penggunaan w tidak berpengaruh terhadap waktu komputasi yang dapat dilihat pada Tabel.4, Tabel.5, Tabel.6. Oleh karena itu, hasil waktu komputasi yang didapat berbeda-beda. Hasil yang bervariasi dikarenakan adanya nilai random yang mempengaruhi perhitungan kecepatan sehingga menghasilkan waktu komputasi yang berbeda-beda

Tabel. 7 Hasil Pengujian Parameter C1 Diturunkan

No	Jumlah Partikel	Berat inersia	C1	C2	Waktu Komputasi	Lokasi	Jarak
1	10	1	2	2	0.1 detik	Klinik Asy-Syifa	0.51
2	10	1	1.5	2	0.1 detik	Klinik Asy-Syifa	0.51
3	10	1	1	2	0.1 detik	Klinik Asy-Syifa	0.51
4	10	1	0.5	2	0.1 detik	Klinik Asy-Syifa	0.51
5	50	1	2	2	0.18 detik	Klinik Asy-Syifa	0.51
6	50	1	1.5	2	0.18 detik	Klinik Asy-Syifa	0.51
7	50	1	1	2	0.18 detik	Klinik Asy-Syifa	0.51
8	50	1	0.5	2	0.18 detik	Klinik Asy-Syifa	0.51
9	100	1	2	2	0.28 detik	Klinik Asy-Syifa	0.51
10	100	1	1.5	2	0.28 detik	Klinik Asy-Syifa	0.51
11	100	1	1	2	0.28 detik	Klinik Asy-Syifa	0.51
12	100	1	0.5	2	0.28 detik	Klinik Asy-Syifa	0.51

Dapat dilihat pada Tabel.7, dari 12 kali percobaan, dengan koordinat awal  $x = -3.800528$  dan  $y = 102.272241$ , dengan jumlah partikel = 10,  $C_1 = (1.5, 1, \text{ dan } 0.5)$ , dan  $C_2 = 2$ , diperoleh waktu komputasi yang sama dengan waktu 0,18 detik. Percobaan dengan jumlah partikel = 50,  $C_1 = (1.5, 1, \text{ dan } 0.5)$ , dan  $C_2 = 2$ , diperoleh waktu komputasi yang sama dengan waktu 0,18 detik. Percobaan dengan jumlah partikel = 50,  $C_1 = (1.5, 1, \text{ dan } 0.5)$ , dan  $C_2 = 2$ , diperoleh waktu komputasi yang sama yaitu 0.28 detik. Lokasi terdekat yang dihasilkan yaitu Klinik Asy-Syifa dengan jarak 0,51 km.

Tabel. 8 Hasil Pengujian Parameter C2 Diturunkan

No	Jumlah Partikel	Berat inersia	C1	C2	Waktu Komputasi	Lokasi Terdekat	Jarak
1	10	1	2	2	0.1 detik	Klinik Asy-Syifa	0.51
2	10	1	2	1.5	0.11 detik	Klinik	0.51

						Asy-Syifa	
3	10	1	2	1	0.11 detik	Klinik Asy-Syifa	0.51
4	10	1	2	0.5	0.11 detik	Klinik Asy-Syifa	0.51
5	50	1	2	2	0.18 detik	Klinik Asy-Syifa	0.51
6	50	1	2	1.5	0.19 detik	Klinik Asy-Syifa	0.51
7	50	1	2	1	0.19 detik	Klinik Asy-Syifa	0.51
8	50	1	2	0.5	0.19 detik	Klinik Asy-Syifa	0.51
9	100	1	2	2	0.27 detik	Klinik Asy-Syifa	0.51
10	100	1	2	1.5	0.29 detik	Klinik Asy-Syifa	0.51
11	100	1	2	1	0.28 detik	Klinik Asy-Syifa	0.51
12	100	1	2	0.5	0.28 detik	Klinik Asy-Syifa	0.51

Pada Tabel.8, dari 12 kali percobaan, dengan koordinat awal  $x = -3.800528$  dan  $y = 102.272241$ , dengan jumlah partikel = 10,  $C_1 = 2$ , dan  $C_2 = (1.5, 1, 0.5)$ , maka diperoleh waktu komputasi yang sama yaitu 0,11 detik. Percobaan dengan jumlah partikel = 50,  $C_1 = 2$ , dan  $C_2 = (1.5, 1, 0.5)$ , maka diperoleh waktu komputasi yang sama yaitu 0,19 detik. Percobaan dengan jumlah partikel = 100,  $C_1 = 2$ , dan  $C_2 = (1.5, 1, 0.5)$ , diperoleh waktu komputasi yang sedikit berbeda yaitu secara berturut-turut, 0,29 detik, 0,28 detik, dan 0,28 detik. Lokasi terdekat yang dihasilkan yaitu Klinik Asy-Syifa dengan jarak 0,51 km.

Berdasarkan teori yang dijelaskan, bahwa penggunaan  $C_2$  yang lebih besar memberikan hasil yang lebih efisien dibandingkan  $C_1$  yang lebih besar[3].  $C_2$  lebih besar memberikan hasil efisien dikarenakan partikel bergerak kearah *global best* sehingga lebih cepat menemukan solusi terbaik. Jika  $C_1$  lebih besar maka partikel akan bergerak kearah *local best*[3]. Hal ini terbukti dengan pengujian yang dilakukan pada Tabel.7 dan Tabel.8 bahwa penggunaan parameter  $C_1$  yang lebih kecil atau dibawah 2, menghasilkan waktu komputasi dengan jumlah partikel = 10, waktunya adalah 0,1 detik, dengan jumlah partikel = 50, waktunya adalah 0,18 detik, dan dengan jumlah partikel = 100, waktunya adalah 0,28 detik. Penggunaan  $C_2$  yang lebih kecil

atau di bawah 2, menghasilkan waktu komputasi dengan jumlah partikel 10 = 0,11 detik, jumlah partikel 50 = 0,19 detik, dan jumlah partikel 100 = 0,28-0,29 detik. Dapat disimpulkan bahwa penggunaan  $C_2$  yang lebih besar atau  $C_1$  yang lebih kecil menghasilkan waktu komputasi / waktu pencarian untuk menemukan solusi terbaik lebih cepat dibandingkan  $C_1$  yang lebih besar atau  $C_2$  yang lebih kecil. Lokasi terdekat yang dihasilkan yaitu Klinik Asy-Syifa dengan jarak 0,51 km.

## VI. PENUTUP

### 6.1 Kesimpulan

Berdasarkan analisis perancangan, implementasi, dan pengujian sistem, maka disimpulkan beberapa hal berikut :

1. Penelitian ini berhasil diimplementasikan pada pencarian lokasi fasilitas kesehatan terdekat menggunakan Algoritma APSO.
2. Penelitian ini telah berhasil merancang dan membangun aplikasi pencarian lokasi fasilitas kesehatan terdekat di Kota Bengkulu berbasis *Web View Android*. Sistem ini dapat digunakan oleh pengguna untuk membantu mencari lokasi fasilitas kesehatan terdekat.
3. Ketetapan parameter partikel = 10, 50, 100,  $w = 1$ ,  $C_1 = 2$  dan  $C_2 = 2$ . Pada pengujian pertama yaitu pengujian partikel 10, 50 dan 100, bahwa semakin kecil jumlah partikel maka waktu komputasinya akan semakin cepat. Sebaliknya, jika jumlah partikel besar maka waktu komputasinya akan lebih lama.
4. Ketetapan parameter partikel = 10, 50, 100,  $w = 1$ ,  $C_1 = 2$  dan  $C_2 = 2$ . Pada pengujian kedua yaitu pengujian parameter berat inersia ( $w$ ) diperoleh hasil penggunaan  $w$  tidak berpengaruh terhadap waktu komputasi, sehingga ketika  $w$  diturunkan dari 1 sampai 0,1 menghasilkan waktu komputasi yang bervariasi. Penggunaan  $w$  berpengaruh untuk meningkatkan performasni PSO yang terbukti dengan menemukan lokasi terdekat yaitu Klinik Asy-Syifa dengan jarak 0,51 km.
5. Ketetapan parameter partikel = 10, 50, 100,  $w = 1$ ,  $C_1 = 2$  dan  $C_2 = 2$ . Pada pengujian ketiga yaitu pengujian parameter  $C_1 = 0.5, 1, 1$  dan  $C_2 = 2$ . Dan sebaliknya  $C_1 = 2$  dan  $C_2 = 0.5, 1, 1$ . Hasil yang diperoleh bahwa penggunaan  $C_2$  yang lebih besar atau  $C_1$  yang lebih kecil menghasilkan waktu komputasi / waktu pencarian untuk menemukan solusi terbaik lebih cepat dibandingkan  $C_1$  yang lebih besar atau  $C_2$  yang lebih kecil. Lokasi terdekat yaitu Klinik Asy-Syifa dengan jarak 0,51 km.

## 6.2 Saran

Berdasarkan analisa perancangan sistem, implementasi, dan pengujian sistem, maka untuk pengembangan penelitian selanjutnya penulis menyarankan hal berikut :

1. Perlu dilakukan pengembangan aplikasi yang dapat mendeteksi posisi pengguna secara otomatis.
2. Perlu dilakukan pengembangan aplikasi yang mengizinkan pengguna untuk melakukan penambahan data, agar data fasilitas kesehatan dapat terus diperbarui.
3. Perlu dilakukan penelitian lanjut mengenai pencarian metode *Particle Swarm Optimization* dengan penambahan parameter lainnya.
4. Perlu dilakukan penelitian untuk kasus ini menggunakan metode pencarian optimasi jenis lainnya, sehingga dapat dilakukan perbandingan hasil yang paling optimal.

## REFERENSI

- [1] Azlina, N., & Ibrahim, Z. (2012). Asynchronous Particle Swarm Optimization for Swarm Robotics. *International Symposium on Robotics and Intelligent Sensors*, 7.
- [2] Cempaka, H. (2015). Penerapan Algoritma Ant Colony System Pada Aplikasi Pencarian Lokasi Tambal Ban Terdekat di Kota Bengkulu Berbasis Android. *Informatika*, 3-4.
- [3] Jatmiko, W., Febrian, A., Jovan, F., Suryana, M. E., Alvisalim, M. S., & Insani, A. (2010). *Swarm Robot dalam Pencarian Sumber Asap*. Jawa Barat: Fakultas Ilmu Komputer Universitas Indonesia.
- [4] Prasetyaningrum, I., Fariza, A., & Fuady, A. (2011). Aplikasi GIS Berbasis J2ME Pencarian Jalur Terpendek Menggunakan Algoritma Particle Swarm Optimization (PSO) Di Kabupaten Bangkalan. *Industrial Electronics Seminar*, 1-3.
- [5] Santoso, B. (2012). Tutorial Particle Swarm Optimization. *Teknik Industri*, 15.
- [6] Suyanto. (2014). *Algoritma Optimasi*. Yogyakarta: Graha Ilmu.



